

## Analisis Aplikasi Algoritma Genetika dalam Proses Desain Struktur Perkerasan

Djunaedi Kosasih<sup>1)</sup>

### Abstrak

*Algoritma genetika pada prinsipnya bermanfaat untuk persoalan yang sulit dipecahkan dengan menggunakan pendekatan deterministik. Meskipun demikian, makalah ini dimaksudkan hanya untuk mengeksplorasi aplikasi algoritma genetika yang menggunakan pendekatan stochastic sebagai metoda alternatif dalam mencari nilai ITP (Index Tebal Perkerasan) dari model empiris yang memerlukan proses konvergensi sederhana. Kemudian, nilai ITP tersebut dibandingkan dengan yang diperoleh melalui teknik interpolasi linier yang sederhana dalam pencarian nilai konvergen. Dua proses kodifikasi solusi (i.e. nilai ITP) yang diperlukan pada algoritma genetika apakah sebagai bilangan biner (kode konvensional) atau bilangan riil (kode non-konvensional) dianalisis secara khusus, mengingat kode bilangan riil seringkali diperlukan untuk aplikasi algoritma genetika dalam bidang rekayasa transportasi. Empat parameter utama dari algoritma genetika, yaitu jumlah populasi, probabilitas pertukaran gen, probabilitas perubahan gen dan fungsi seleksi, yang dianggap paling sesuai untuk pencarian nilai ITP diusulkan dalam makalah ini.*

**Kata-kata Kunci :** Algoritma genetika, index tebal perkerasan, proses konvergensi.

### Abstract

*Genetic algorithm basically is a useful tool for problems that are difficult to solve by using deterministic approaches. Despite of this, this paper is intended only to explore genetic algorithm which is based on the principle of stochastic approaches as an alternative method to calculate the structural number of a pavement from an empirical model requiring simple convergency processes. The resulting pavement structural number was then compared with that obtained from a simple linear interpolation technique in seeking for a convergence. Two codification processes of solution (i.e. structural number) performed in genetic algorithm by using binary number (conventional code) or real number (non-conventional code) were specifically studied, since the real number code is frequently used for the application of genetic algorithm in the field of transportation engineering. Four main parameters needed for genetic algorithm, i.e. population size, probability of crossover, probability of mutation and selection function, that were found to be the most appropriate for searching pavement structural number are proposed in this paper.*

**Keywords :** Genetic algorithm, structural number, convergency process.

## 1. Pendahuluan

Algoritma genetika pada prinsipnya bermanfaat untuk persoalan yang sulit dipecahkan dengan menggunakan pendekatan deterministik. Meskipun demikian, makalah ini dimaksudkan hanya untuk mengeksplorasi aplikasi algoritma genetika yang menggunakan pendekatan *stochastic* sebagai metoda alternatif dalam mencari nilai ITP (Index Tebal Perkerasan) dari model empiris yang memerlukan proses konvergensi yang sederhana.

Nilai ITP menurut metoda AASHTO [1993] untuk struktur perkerasan lentur dapat dihitung dari model

empiris, sebagai berikut:

$$\log(N) = Z_R \times S_o + 9.36 \times \log\left(\frac{ITP}{2.54} + 1\right) - 3.25571$$

$$+ \frac{\log\left[\frac{\Delta IP}{4.2 - 1.5}\right]}{0.40 + \frac{1094}{\left(\frac{ITP}{2.54} + 1\right)^{5.19}}} + 2.32 \times \log(M_R) \quad (1)$$

dimana:

$N$  = jumlah total lintasan sumbu standar ekuivalen pada lajur desain selama masa layan rencana struktur perkerasan (ss)

1. Staf Pengajar Departemen Teknik Sipil, FTSP-ITB, dan Universitas Tarumanegara, Jakarta

**Catatan :** Usulan makalah dikirimkan pada 16 Juni 2005 dan dinilai oleh peer reviewer pada tanggal 20 Juni 2005 - 06 Juli 2005. Revisi penulisan dilakukan antara tanggal 06 Juli 2005 hingga 11 Juli 2005.

- $Z_R$  = konstanta normal pada probabilitas  $R\%$   
 $S_o$  = deviasi standar dari nilai  $\log(N)$   
 $ITP$  = Index Tebal Perkerasan (cm)  
 $\Delta IP$  = rentang nilai kondisi struktur perkerasan (Index Permukaan) dari awal sampai akhir masa layan rencana struktur perkerasan  
 $M_R$  = modulus resilient tanah dasar (MPa)

Dalam proses desain struktur perkerasan, ada lima data input (*i.e.* nilai  $N$ ,  $Z_R$ ,  $S_o$ ,  $\Delta IP$  dan  $M_R$ ) yang perlu ditetapkan untuk menghitung nilai  $ITP$  sebagai solusi. Jelaslah, bahwa perhitungan nilai  $ITP$  dari **Persamaan (1)** dapat dilakukan melalui proses konvergensi dengan menggunakan teknik interpolasi linier secara iteratif, seperti diuraikan pada **Bab 2**.

Proses konvergensi untuk menyelesaikan **Persamaan (1)** didasarkan pada nilai *deviasi* yang dihitung dari persamaan berikut:

$$\text{deviasi} = \text{nilai kanan} - \text{nilai kiri} \quad (2)$$

dimana: nilai kiri =  $\log(N)$

$$\begin{aligned} \text{nilai kanan} = & Z_R \times S_o + 9.36 \times \log\left(\frac{ITP}{2.54} + 1\right) - 3.25571 \\ & + \frac{\log\left[\frac{\Delta IP}{4.2 - 1.5}\right]}{0.40 + \frac{1094}{\left(\frac{ITP}{2.54} + 1\right)^{5.19}}} + 2.32 \times \log(M_R) \end{aligned}$$

Jika nilai  $ITP$  yang dimasukkan ke dalam **Persamaan (2)** lebih besar dari pada solusi yang benar, maka nilai *deviasi* akan positif. Sedangkan, jika nilai  $ITP$  lebih kecil, maka nilai *deviasi* akan menjadi negatif. Nilai  $ITP$  yang merupakan solusi yang tepat dari **Persamaan (2)** yang juga merupakan solusi dari **Persamaan (1)** akan diperoleh melalui proses interpolasi linier, jika nilai *deviasi* = 0.

Akan tetapi, untuk keperluan desain praktis, nilai  $ITP$  cukup memiliki ketelitian 2 desimal. Sehingga, solusi **Persamaan (1)** yang diperoleh harus memenuhi dua ketentuan berikut:

- nilai  $ITP$  terkecil dengan ketelitian 2 desimal, dan
- nilai *deviasi*  $\geq 0$

**Tabel 1.** Rentang nilai data input yang diijinkan

No	Tipe Data Input	Rentang Nilai Data	(unit)
1	Beban lalu lintas, $N$	0.10 ÷ 100.00	(jutass)
2	Probabilitas, $R$	50.00 ÷ 99.99	(%)
3	Konstanta normal, $Z_R$	0.000 ÷ -3.750	(-)
4	Deviasi standar dari nilai $\log(N)$ , $S_o$	0.3 ÷ 0.5	(-)
5	Nilai kondisi struktur perkerasan, $\Delta IP$	0.5 ÷ 3.0	(-)
6	Modulus resilient tanah dasar, $M_R$	20.0 ÷ 200	(MPa)

Dua solusi alternatif berdasarkan algoritma genetika yang dikembangkan dalam studi ini yang masing-masing dinyatakan sebagai kode bilangan biner dan kode bilangan riil diuraikan pada **Bab 3** dan **4**. Nilai *deviasi* dari **Persamaan (2)** juga merupakan nilai kecocokan yang digunakan dalam proses evaluasi.

Rentang nilai  $ITP$  yang mungkin digunakan dalam proses desain praktis harus diketahui terlebih dahulu, agar kodifikasi solusi pada algoritma genetika dapat dilakukan. Untuk itu, rentang nilai dari setiap data input yang diijinkan perlu ditetapkan, seperti terlihat pada **Tabel 1**. Substitusi semua rentang nilai data input dari **Tabel 1** ke dalam **Persamaan (1)** dihasilkan rentang nilai  $ITP$  sebesar  $2.67 \div 45.41$ .

Persoalan yang dihadapi di sini pada hakekatnya menghasilkan solusi eksak. Dilain pihak, algoritma genetika menggunakan pendekatan *stochastic* dalam mencari solusi terbaik. Oleh karena itu, jumlah regenerasi yang dilakukan pada algoritma genetika merupakan ketentuan tambahan yang perlu diperhatikan secara khusus. Jumlah regenerasi tidak boleh terlalu sedikit, karena proses konvergensi mungkin masih belum selesai; dan juga, tidak boleh terlalu banyak, karena waktu pemrosesan komputer menjadi lama.

Untuk menghindari kegagalan dalam memperoleh nilai eksak, ketentuan tambahan berikut diperlukan sebagai uji terminasi dari setiap proses regenerasi. Dengan ketentuan tambahan ini, jumlah regenerasi yang dilakukan untuk menghasilkan solusi eksak dapat ditentukan secara otomatis oleh program.

- solusi terbaik pada setiap proses regenerasi ( $ITP_{best}$ ) yang dicari berdasarkan dua ketentuan solusi seperti yang telah disebutkan di atas, dan
- nilai  $\text{deviasi}_{(ITP_{best} - 0.01)} < 0$

Selain itu, dalam makalah ini diusulkan empat parameter utama dari algoritma genetika, yaitu jumlah populasi, probabilitas pertukaran *gen*, probabilitas perubahan *gen* dan fungsi seleksi, yang dianggap paling sesuai untuk pencarian nilai  $ITP$ . Sebagai pembandingan, dari studi terdahulu diketahui bahwa, untuk persoalan pencarian nilai fungsi sinus maksimum dengan kodifikasi solusi sebagai bilangan biner, hanya probabilitas perubahan *gen* dan fungsi seleksi yang mempengaruhi solusi optimum secara signifikan [Kosasih, et.al., 2005].

## 2. Proses Konvergensi dengan Teknik Interpolasi Linier

Perhitungan nilai  $ITP$  melalui proses konvergensi dengan teknik interpolasi linier dilakukan secara coba-

coba. Ada enam langkah iteratif yang harus dilakukan, yaitu:

1. Coba nilai  $ITP_1$ , sebagai perkiraan awal dapat digunakan nilai batas bawah 2.67, dan substitusikan ke **Persamaan (2)** untuk mendapatkan nilai  $deviasi_1$  yang negatif
2. Kemudian, coba nilai  $ITP_2$ , sebagai perkiraan awal juga dapat digunakan nilai batas atas 45.41, dan substitusikan lagi ke **Persamaan (2)**, sehingga nilai  $deviasi_2$  adalah positif; jadi, nilai  $deviasi_2 >$  nilai  $deviasi_1$
3. Hitung nilai  $ITP_i$  dengan teknik interpolasi linier, seperti diilustrasikan pada **Gambar 1**, berdasarkan persamaan berikut:

$$ITP_i = ITP_1 + \frac{0 - deviasi_1}{deviasi_2 - deviasi_1} (ITP_2 - ITP_1) \quad (3)$$

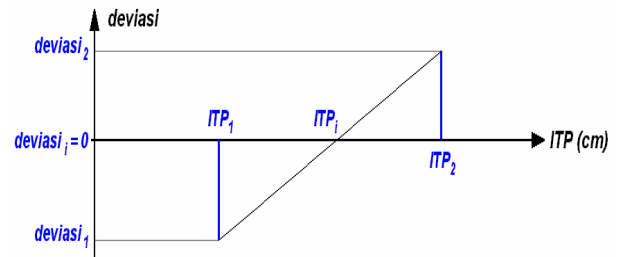
4. Hitung nilai  $deviasi_i$  dengan mensubstitusikan nilai  $ITP_i$  tersebut di atas ke **Persamaan (2)**
5. Jika nilai  $deviasi_i < 0$ , maka ganti nilai  $ITP_1$  dan  $deviasi_1$  dengan nilai  $ITP_i$  dan  $deviasi_i$  atau  
jika nilai  $deviasi_i \geq 0$ , maka ganti nilai  $ITP_2$  dan  $deviasi_2$  dengan nilai  $ITP_i$  dan  $deviasi_i$
6. Jika nilai  $(ITP_2 - ITP_1) \leq 0.01$  dan nilai  $deviasi_2 \geq 0$ , maka proses iterasi selesai dengan konvergensi terjadi pada nilai  $ITP_2$  (sebagai solusi); jika tidak, proses iterasi kembali ke langkah no 3.

### 3. Algoritma Genetika dengan Pengkodean Solusi sebagai Bilangan Biner

Urutan proses dari algoritma genetika dengan pengkodean solusi sebagai bilangan biner diperlihatkan dengan menggunakan contoh sederhana oleh Gen, et.al. [1997] dan Kosasih, et.al. [2005]. Berikut diuraikan secara singkat kelima tahapan proses yang dilakukan.

#### 3.1 Kodifikasi solusi dan pembentukan generasi awal

Kodifikasi solusi sebagai bilangan biner pada algoritma genetika merupakan cara yang paling umum digunakan. Untuk contoh *kasus* yang sedang dikerjakan, karena nilai  $ITP$  sebagai solusi yang dicari adalah dua desimal, maka kode bilangan biner yang digunakan harus dapat mewakili rentang nilai antara



**Gambar 1. Ilustrasi perhitungan nilai  $ITP$  dengan teknik interpolasi linier**

$267 \div 4541$  (13 bit = 8191). Nilai desimal 8191 mewakili nilai  $ITP = 45,41$  cm dan nilai desimal 0 mewakili nilai  $ITP = 2,67$  cm. Nilai  $ITP$  antara dihitung secara proporsional.

Setiap *bit* merupakan *gen* yang membentuk individu. Dengan demikian, masing-masing individu memiliki 13 *gen* sebagai solusi. Misalkan, salah satu individu yang terbentuk secara random adalah 0010111101001 (= 1513). Nilai desimal 1513 ini dikonversikan ke dalam nilai  $ITP$  secara proporsional menjadi:

$$ITP = 2.67 + \frac{1513}{8191} * (45.41 - 2.67) = 10.57 \text{ cm}$$

Generasi awal dibentuk dengan jumlah individu (populasi) yang biasanya ditetapkan antara  $10 \div 100$ , dengan jumlah *gen* per individu terdiri dari bilangan random sebesar 13 *bit*. Secara umum, makin besar populasi yang dibentuk, maka akan makin besar pula kemungkinan solusi optimum yang dapat dihasilkan dari algoritma genetika. Akan tetapi, sebagai konsekuensi, waktu pemrosesan komputer akan menjadi lebih lama.

#### 3.2 Proses pertukaran *gen*

Pertukaran *gen* antara dua individu yang dinyatakan sebagai kode bilangan biner dapat dilakukan dengan menggunakan operator genetika konvensional. Kedua individu yang mengalami proses pertukaran *gen* ditentukan secara random. Hasil yang diperoleh adalah dua individu baru sebagai turunannya.

Cara yang mudah untuk melakukan pertukaran *gen* adalah berdasarkan titik potong yang juga ditentukan secara random. Potongan *gen* sebelah kiri titik potong dari satu individu induk digabungkan dengan potongan *gen* sebelah kanan titik potong dari individu induk lainnya.

Proses pertukaran *gen* untuk dua individu induk yang terpilih diilustrasikan pada **Tabel 2**, dimana titik potong dimisalkan terjadi pada *gen* ke lima. Seperti terlihat, dua individu turunan yang dihasilkan memiliki nilai  $ITP$  yang berbeda. Selanjutnya, jumlah

individu yang mengalami proses pertukaran *gen* pada satu generasi ditentukan secara random berdasarkan tingkat probabilitas pertukaran *gen* yang diijinkan. Jika tingkat probabilitas pertukaran *gen* yang diijinkan adalah 25%, maka untuk generasi dengan populasi yang terdiri dari 10 individu, jumlah individu yang akan mengalami proses pertukaran *gen* kurang lebih adalah 2 atau 3.

### 3.3 Proses perubahan *gen*

Perubahan *gen* merupakan operator genetika ke dua dan hanya bekerja pada beberapa *gen* dari individu yang melakukan penyesuaian diri terhadap kondisi lingkungan sekitar. Proses perubahan *gen* terjadi agar makhluk hidup dapat terus bertahan hidup dengan kualitas yang lebih baik.

Pada algoritma genetika, proses perubahan *gen* yang menghasilkan *gen* yang lebih baik dapat membuat individu tetap bertahan dalam proses seleksi dan diharapkan akan dapat makin mendekati solusi optimum. Sebaliknya, proses perubahan *gen* yang menghasilkan *gen* yang lebih buruk dapat membuat individu tersebut tereliminasi dalam proses seleksi.

Untuk individu yang dinyatakan dengan kode bilangan biner, proses perubahan *gen* juga dilakukan dengan menggunakan operator genetika konvensional, yaitu mengubah nilai 0 menjadi 1, atau sebaliknya (lihat **Tabel 3**). Jumlah *gen* yang mengalami proses perubahan *gen* pada satu generasi ditentukan secara random berdasarkan tingkat probabilitas perubahan *gen* yang diijinkan. Untuk contoh kasus pencarian nilai *ITP* dengan populasi misalnya sebesar 10x13 *gen*, jika tingkat probabilitas yang diijinkan adalah 10%, maka jumlah *gen* yang akan mengalami proses perubahan *gen* pada satu generasi kurang lebih 13.

**Tabel 2. Contoh proses pertukaran *gen***

2 individu yang mengalami proses pertukaran <i>gen</i>		
kode bilangan biner	nilai desimal	nilai ITP
0010 <u>111101001</u>	1513	10.56
1011 <u>101111010</u>	6010	34.03



2 individu turunan yang terbentuk		
kode bilangan biner	nilai desimal	nilai ITP
0010 <u>101111010</u>	1402	9.99
1011 <u>111101001</u>	6121	34.61

**Tabel 3. Contoh proses perubahan *gen***

2 individu yang mengalami proses perubahan <i>gen</i>		
kode bilangan biner	nilai desimal	nilai ITP
<u>1</u> 011111101001	6121	34.61



2 individu turunan yang terbentuk		
kode bilangan biner	nilai desimal	nilai ITP
<u>0</u> 011111101001	2025	13.24

### 3.4 Proses evaluasi

Proses evaluasi pada algoritma genetika dimaksudkan untuk menghitung nilai kecocokan terhadap solusi optimum dari setiap individu turunan yang telah mengalami proses pertukaran *gen* dan perubahan *gen*. Untuk contoh kasus pencarian nilai *ITP*, nilai kecocokan yang dimaksud adalah nilai *deviasi* yang dihitung dari **Persamaan (2)** dengan mensubstitusikan nilai *ITP* yang mewakili setiap individu turunan tersebut.

Dalam hal ini, individu terbaik adalah individu yang memiliki nilai *deviasi* positif terkecil. Individu turunan yang dihasilkan tidak harus selalu lebih baik dari pada individu induk.

### 3.5 Proses seleksi

Proses seleksi dilakukan untuk memilih individu induk dan individu turunan berdasarkan nilai kecocokan yang diperoleh di atas untuk membentuk generasi baru yang lebih baik ke arah solusi optimum yang dicari. Ada tiga ketentuan dasar yang harus dipertimbangkan dalam melakukan proses seleksi, yaitu:

- Jumlah populasi pada setiap generasi baru harus selalu dipertahankan tetap (*n*).
- Fungsi seleksi yang cocok harus dipilih sesuai dengan jenis aplikasinya. Empat alternatif fungsi seleksi yang disediakan pada program Genetika [Kosasih, 2005], termasuk:
  - Fungsi seleksi random; generasi baru dipilih secara random berdasarkan kurva distribusi kumulatif nilai kecocokan, baik yang berasal dari individu induk, maupun dari individu turunan dengan prosedur, sbb.:

- i) hitung probabilitas nilai kecocokan,

$$P_i = (\sum_{j=1}^i deviasi_j) / (\sum_{j=1}^{2n} deviasi_j)$$

- ii) ambil nilai random,  $P_{random} (= 0 \div 1)$   
 iii) pilih individu  $i$  yang memenuhi kondisi ( $P_i < P_{random} \leq P_{i+1}$ )  
 iv) ulangi langkah ii) dan iii) untuk  $n$  individu yang membentuk generasi baru

2. Fungsi seleksi random terkoreksi; sama seperti alternatif fungsi 1 di atas tetapi individu terbaik dari individu induk atau individu turunan harus selalu disertakan dalam generasi baru.
3. Fungsi seleksi turunan terkoreksi; semua individu turunan dipilih sebagai generasi baru, kecuali individu terburuk yang digantikan oleh individu terbaik dari generasi induk; dan untuk menghindari duplikasi, individu pengganti harus lebih baik dibandingkan dengan yang terbaik dari individu turunan.
4. Fungsi seleksi unggul; semua individu yang memiliki nilai kecocokan terbaik dipilih dari gabungan antara individu induk dan individu turunan sebagai generasi baru.

- c. Duplikasi individu pada generasi baru harus dapat dicegah untuk menghindari proses pencarian terperangkap pada solusi optimum lokal. Disamping itu, nilai fungsi individu yang saling berdekatan seharusnya juga tidak disukai karena akan mempersempit ruang pencarian.

Sejauh ini, satu siklus regenerasi telah diuraikan secara singkat. Individu terbaik dari generasi baru yang dihasilkan mungkin masih belum merupakan solusi optimum yang dicari. Proses regenerasi kemudian terus berlanjut sampai solusi optimum dapat diperoleh sesuai dengan ketentuan tambahan yang telah dijelaskan di atas.

#### 4. Algoritma Genetika dengan Pengkodean Solusi sebagai Bilangan Riil

Algoritma genetika dengan pengkodean solusi sebagai bilangan riil telah banyak diaplikasikan. Kekhususan dari kode bilangan riil adalah dalam hal operator genetika yang digunakan. Operator non-konvensional,

baik untuk pertukaran gen, maupun untuk perubahan gen, harus dipilih secara tepat sesuai dengan aplikasi yang dikerjakan.

Kelima tahapan proses dari algoritma genetika untuk solusi yang dinyatakan sebagai bilangan riil diuraikan secara rinci berikut ini. Satu siklus regenerasi mulai dari generasi awal sampai pada generasi ke dua dijelaskan dengan menggunakan contoh yang lengkap.

##### 4.1 Kodifikasi solusi dan pembentukan generasi awal

Kodifikasi solusi sebagai bilangan riil pada dasarnya dapat dilakukan secara langsung berdasarkan rentang nilai solusi yang diijinkan. Sebagai contoh, 10 individu yang mungkin terbentuk secara random untuk menjadi generasi awal, beserta dengan nilai *deviasi* masing-masing yang dihitung dari **Persamaan (2)**, adalah sebagai berikut: (**Tabel 4**)

Dari ketentuan yang telah ditetapkan di atas, individu 8 merupakan solusi yang terbaik dan individu 4 merupakan solusi yang terburuk dari generasi awal ini.

##### 4.2 Proses pertukaran gen

Proses pertukaran *gen* antara dua individu induk yang dinyatakan sebagai kode bilangan riil dapat dilakukan dengan menggunakan operator genetika non-konvensional. Ada dua jenis operator genetika non-konvensional yang dikenal, yaitu operator genetika aritmatik dan operator genetika direksional [Gen, et.al., 1997].

Operator genetika aritmatik pada dasarnya tidak selalu dapat memberikan kepastian bahwa individu turunan yang dihasilkan menjadi lebih baik dibandingkan dengan individu induknya. Untuk contoh kasus pencarian nilai *ITP*, ketentuan tentang solusi optimum dapat disertakan ke dalam operator genetika direksional, sehingga individu turunan yang dihasilkan diharapkan akan selalu lebih baik.

Kedua individu yang mengalami proses pertukaran *gen* tetap perlu ditentukan secara random. Akan tetapi, hasil yang diperoleh dalam hal ini hanyalah satu individu baru saja sebagai turunannya, seperti terlihat pada **Persamaan (4)** khusus untuk nilai *deviasi* yang positif. Untuk nilai *deviasi* yang negatif atau yang

**Tabel 4. Contoh generasi awal yang terbentuk**

No Individu	1	2	3	4	5	6	7	8	9	10
<i>ITP (cm)</i>	24.16	37.56	12.62	41.17	24.88	22.15	39.02	18.87	29.96	4.13
<i>deviasi</i>	2.1705	3.8163	-0.0083	4.1663	2.2776	1.8566	3.9615	1.2906	2.9641	-2.8714

berbeda tanda, **Persamaan (4)** dapat disesuaikan seperlunya. Setelah proses pertukaran *gen* dilakukan, nilai  $ITP'$  seharusnya akan lebih baik daripada, atau setidaknya sama dengan, nilai  $ITP_1$  atau nilai  $ITP_2$

$$ITP' = ITP_1 + R * (ITP_1 - ITP_2)$$

→ untuk  $ITP_1 < ITP_2$  dan nilai deviasi  $> 0.0$

atau

$$ITP' = ITP_2 + R * (ITP_2 - ITP_1) \text{ dimana:}$$

→ untuk  $ITP_1 > ITP_2$  dan nilai deviasi  $> 0.0$

$$ITP' = \text{individu turunan} \quad (4)$$

$ITP_1, ITP_2$  = dua individu induk yang mengalami proses pertukaran *gen*

$$R = \text{bilangan random } (= 0 \div 1)$$

Misalkan, untuk tingkat probabilitas pertukaran *gen* sebesar 25%, individu 1 dan individu 7 terpilih secara random untuk mengalami proses pertukaran *gen* pada generasi awal ini. Nilai  $R$  pada contoh ini adalah 0.70. Individu turunan yang dihasilkan berdasarkan **Persamaan (4)** adalah 13.76.

Kesepuluh individu setelah mengalami proses pertukaran *gen* diperlihatkan pada **Tabel 5**. Terlihat bahwa individu 7 menguat dengan nilai *deviasi* turun dari 3.9615 (lihat **Tabel 4**) menjadi 0.2512; dan, individu 7 juga lebih baik dari individu 1 (sebagai individu induk ke dua) yang memiliki nilai *deviasi* 2.1705.

### 4.3 Proses perubahan *gen*

Proses perubahan *gen* pada individu turunan yang dinyatakan sebagai kode bilangan riil juga dilakukan dengan menggunakan operator genetika non-konvensional. Seperti halnya pada proses pertukaran *gen*, untuk contoh kasus pencarian nilai  $ITP$ , operator genetika direksional lebih cocok digunakan, seperti terlihat pada **Persamaan (5)**. Pertukaran *gen* ini dilakukan relatif terhadap nilai rata-rata dari semua individu yang ada.

dimana:

$$ITP' = ITP_1 - R * deviasi_i * |gradient|^{ITP'} \quad (5)$$

**Tabel 5. Contoh proses pertukaran *gen***

No Individu	1	2	3	4	5	6	7	8	9	10
<i>ITP (cm)</i>	24.16	37.56	12.62	41.17	24.88	22.15	13.76	18.87	29.96	4.13
<i>deviasi</i>	2.1705	3.8163	-0.0083	4.1663	2.2776	1.8566	0.2512	1.2906	2.9641	-2.8714

= individu turunan

$ITP_1$  = individu induk yang mengalami proses perubahan *gen*

$R$  = bilangan random  $(= 0 \div 1)$

*deviasi* = nilai deviasi dari Pers (2) untuk nilai  $ITP_1$

$$gradient = \frac{\frac{\sum_{i=1}^{jml \text{ populasi}, n} ITP_i}{n} - ITP_1}{\frac{\sum_{i=1}^{jml \text{ populasi}, n} deviasi_i}{n} - deviasi_1}$$

**Persamaan (5)** merupakan rumus interpolasi dinamis, yang serupa dengan **Persamaan (3)**. Sebagai model *stochastic*, **Persamaan (5)** memiliki variabel tambahan  $R$  yang harus ditentukan secara random.

Pada generasi awal yang sedang dianalisis, untuk tingkat probabilitas 10%, individu 1, 2, 4 dan 10 terpilih secara random untuk mengalami proses pertukaran *gen*. Pada contoh ini, nilai  $R$  adalah 0.62.

Keempat individu turunan yang dihasilkan beserta dengan individu-individu lainnya diperlihatkan pada **Tabel 6**. Setelah mengalami proses pertukaran *gen*, keempat individu menjadi lebih baik dengan nilai *deviasi* yang mendekati nol (bandingkan dengan **Tabel 5**).

### 4.4 Proses evaluasi

Proses evaluasi untuk solusi yang menggunakan kode bilangan riil pada dasarnya tidak berbeda dengan yang menggunakan kode bilangan biner (lihat **Butir 3.4**). Sebagai contoh, nilai *deviasi* sebagai nilai kecocokan yang dihitung berdasarkan **Persamaan (2)** telah turut disertakan pada **Tabel 6**, dan beberapa tabel sebelumnya.

### 4.5 Proses seleksi

Proses seleksi untuk solusi yang menggunakan kode bilangan riil juga tidak berbeda dengan yang menggunakan kode bilangan biner (lihat **Butir 3.5**). Sebagai contoh, untuk fungsi seleksi unggul, generasi ke dua dapat dihasilkan dengan menyeleksi individu



Tabel 6. Contoh proses perubahan *gen*

No Individu	1	2	3	4	5	6	7	8	9	10
<i>ITP (cm)</i>	21.29	22.00	12.62	22.87	24.88	22.15	13.76	18.87	29.96	11.63
<i>deviasi</i>	1.7154	1.8318	-0.0083	1.9714	2.2776	1.8566	0.2512	1.2906	2.9641	-0.2432

Tabel 7. Contoh generasi kedua yang dihasilkan

No Individu	1	2	3	4	5	6	7	8	9	10
<i>ITP (cm)</i>	21.29	22.00	12.62	22.87	24.88	22.15	13.76	18.87	11.63	24.16
<i>deviasi</i>	1.7154	1.8318	-0.0083	1.9714	2.2776	1.8566	0.2512	1.2906	-0.2432	2.1705

yang terbaik dari individu induk dan individu turunan. Hasil yang diperoleh diperlihatkan pada **Tabel 7**, dimana individu 1 ÷ 9 dipilih dari individu turunan (lihat **Tabel 6**) dan hanya individu 10 dipilih dari individu induk (lihat **Tabel 4**).

## 5. Analisis Hasil Perhitungan Nilai *ITP*

Untuk menguji efektifitas dari algoritma genetika seperti yang telah diuraikan di atas digunakan data input, sebagai berikut:

- Beban lalu lintas,  $N = 10.00$  (juta ss)
- Tingkat probabilitas,  $R = 90.00$  (%)  $\rightarrow Z_R = -1.282$
- Deviasi standar untuk nilai  $\log(N)$ ,  $S_o = 0.35$
- Nilai kondisi struktur perkerasan,  $\Delta IP = 1.50$
- Modulus resilient tanah dasar,  $M_R = 50.00$  (MPa)

Tampilan program Genetika untuk pencarian nilai *ITP* diperlihatkan pada **Gambar 2**. Proses regenerasi dapat dilakukan sampai 5000 kali. Akan tetapi, dengan ketentuan tambahan untuk uji terminasi, proses regenerasi dapat langsung dihentikan pada saat solusi optimum telah diperoleh. Dalam proses analisis selanjutnya, jumlah regenerasi dijadikan sebagai indikator tentang efektifitas dari algoritma genetika yang dilakukan.

Tiga parameter utama lainnya dari algoritma genetika, yaitu jumlah populasi, probabilitas *crossover* (pertukaran *gen*) dan probabilitas *mutation* (perubahan *gen*) dapat diubah untuk kepentingan riset. Representasi *gen* apakah sebagai kode bilangan biner atau sebagai kode bilangan riil juga dapat dipilih secara bergantian.

### 5.1 Perhitungan melalui proses konvergensi dengan teknik interpolasi linier

Proses konvergensi nilai *ITP* dengan menggunakan teknik interpolasi linier disediakan pada program Genetika hanya sebagai pembandingan. Proses iterasi yang dilakukan berdasarkan **Persamaan (3)** disajikan

secara lengkap pada **Tabel 8**.

Jumlah iterasi yang diperlukan untuk contoh ini adalah 10 kali, dimana iterasi ke-11 merupakan uji terminasi. Nilai *ITP* yang diperoleh sama dengan 12.66 cm.

### 5.2 Perhitungan dengan algoritma genetika – kode solusi sebagai bilangan biner

**Gambar 3** memperlihatkan pengaruh dari empat parameter utama algoritma genetika terhadap jumlah regenerasi yang diperlukan dalam memperoleh nilai *ITP* untuk kodifikasi solusi sebagai bilangan biner. Untuk menggambarkan keempat faktor pengaruh tersebut digunakan nilai parameter berikut sebagai acuan.

Jumlah populasi = 10

Tingkat probabilitas pertukaran *gen* = 25 %

Tingkat probabilitas perubahan *gen* = 10 %

Kriteria seleksi : fungsi seleksi unggul (alternatif 4)

Hasil yang diperlihatkan pada **Gambar 3** untuk setiap variasi parameter utama algoritma genetika dalam beberapa (5 ÷ 10) kali pengoperasian terlihat selalu

Tabel 8. Proses konvergensi nilai *ITP* dengan teknik interpolasi linier

Iterasi Ke-	<i>ITP</i> <sub>1</sub> (cm)	<i>deviasi</i> <sub>1</sub>	<i>ITP</i> <sub>2</sub> (cm)	<i>deviasi</i> <sub>2</sub>	<i>ITP</i> <sub>i</sub> (cm)	<i>deviasi</i> <sub>i</sub>
1	267	-3.8520	45.41	4.5424	22.28	1.8780
2	267	-3.8520	22.28	1.8780	15.85	0.7022
3	267	-3.8520	15.85	0.7022	13.82	0.2653
4	267	-3.8520	13.82	0.2653	13.10	0.1030
5	267	-3.8520	13.10	0.1030	12.83	0.0406
6	267	-3.8520	12.83	0.0406	12.73	0.0161
7	267	-3.8520	12.73	0.0161	12.68	0.0064
8	267	-3.8520	12.68	0.0064	12.67	0.0025
9	267	-3.8520	12.67	0.0025	12.66	0.0010
10	267	-3.8520	12.66	0.0010	12.66	0.0004
11			12.65	-0.0013		

**Examples of Genetic Algorithm Processes**

Program ini mengilustrasikan contoh proses Genetic Algorithm untuk mencari nilai fungsi logaritmis sederhana (nilai ITP) - yang kemudian hasilnya dibandingkan dengan hasil perhitungan konvergensi dengan teknik interpolasi linier. Fungsi logaritmis yang dimaksud adalah:

$$\log(N) = Z_s \times S_s + 9.36 \times \log\left(\frac{ITP}{2.54} + 1\right) - 3.25571 + \frac{\log\left[\frac{\Delta ITP}{4.2 - 1.5}\right]}{0.40 + \frac{1094}{\left(\frac{ITP}{2.54} + 1\right)^{2.15}}} + 2.32 \times \log(M_s)$$

**L = R**

**Representasi Gen:** Bilangan Biner

Jumlah Populasi = 10 (10 - 100)      Jumlah Gen = 13

Jumlah Re-Generasi = 5000      Prob Crossover = 0.25 (0.00-1.00)      Prob Mutation = 0.10 (0.00-0.50)

**Data Input:** N (mill.ESA) = 10.0 (0.1 - 100)  
 R (%) = 90.00 (50 - 99.99)  
 So = 0.35 (0.30 - 0.50)  
 IPo - IPt = 1.5 (0.5 - 3.0)  
 Mr (MPa) = 50 (20 - 200)

**Monitoring generasi yang dihasilkan:**

24.16
37.56
12.62
41.17
24.88
22.15
39.02
18.87
29.96
4.13

**Generasi Awal:** 18.870

**Generasi Berikut:** 1      **Generasi No** 1      **ITP (cm)** 18.870000      **Generasi Terbaik No** 1      **ITP (cm)** 18.870      **Deviasi (=R-L)** 1.290581

**Generasi Terakhir:** Proses konvergensi dengan teknik interpolasi linier, ITP (cm) = 12.66 (0.00)

Ref: AASHTO (1996), "AASHTO Guide for Design of Pavement Structures", Washington DC, USA. < Developed by Dr. D. Kasasih >

Gambar 2. Contoh data input (default) pada program Genetika dengan hasil generasi awal

bervariasi. Hal ini pada prinsipnya memang seharusnya terjadi dalam proses pencarian nilai optimum dengan menggunakan algoritma genetika yang didasarkan pada pendekatan *stochastic*. Meskipun demikian, untuk kodifikasi solusi sebagai bilangan biner, nilai *ITP* yang dicari pada umumnya dapat diperoleh, kecuali jika probabilitas perubahan *gen* sama dengan nol. Tanpa perubahan *gen*, proses pencarian nilai *ITP* seringkali tidak dapat memberikan ketelitian yang diharapkan (2 desimal).

Pengaruh dari variasi parameter utama algoritma genetika terhadap jumlah regenerasi yang diperlukan untuk menghasilkan solusi optimum cukup berbeda. Probabilitas perubahan *gen* dan fungsi seleksi memberikan pengaruh yang sangat dominan. Sedangkan, pengaruh dari jumlah populasi dan probabilitas pertukaran *gen* relatif tidak begitu signifikan dan cenderung variatif. Hasil ini ternyata konsisten dengan hasil studi terdahulu seperti yang telah diungkapkan sebelumnya (lihat **Bab I**).

Untuk kodifikasi solusi sebagai bilangan biner, jumlah regenerasi yang efisien adalah sekitar 25 kali. Nilai parameter utama algoritma genetika yang dapat menjadikan proses pencarian nilai *ITP* efisien adalah:

Jumlah populasi = 40  
 Tingkat probabilitas pertukaran *gen* = 25 %

Tingkat probabilitas perubahan *gen* = 20 %

Kriteria seleksi : fungsi seleksi unggul (alternatif 4)

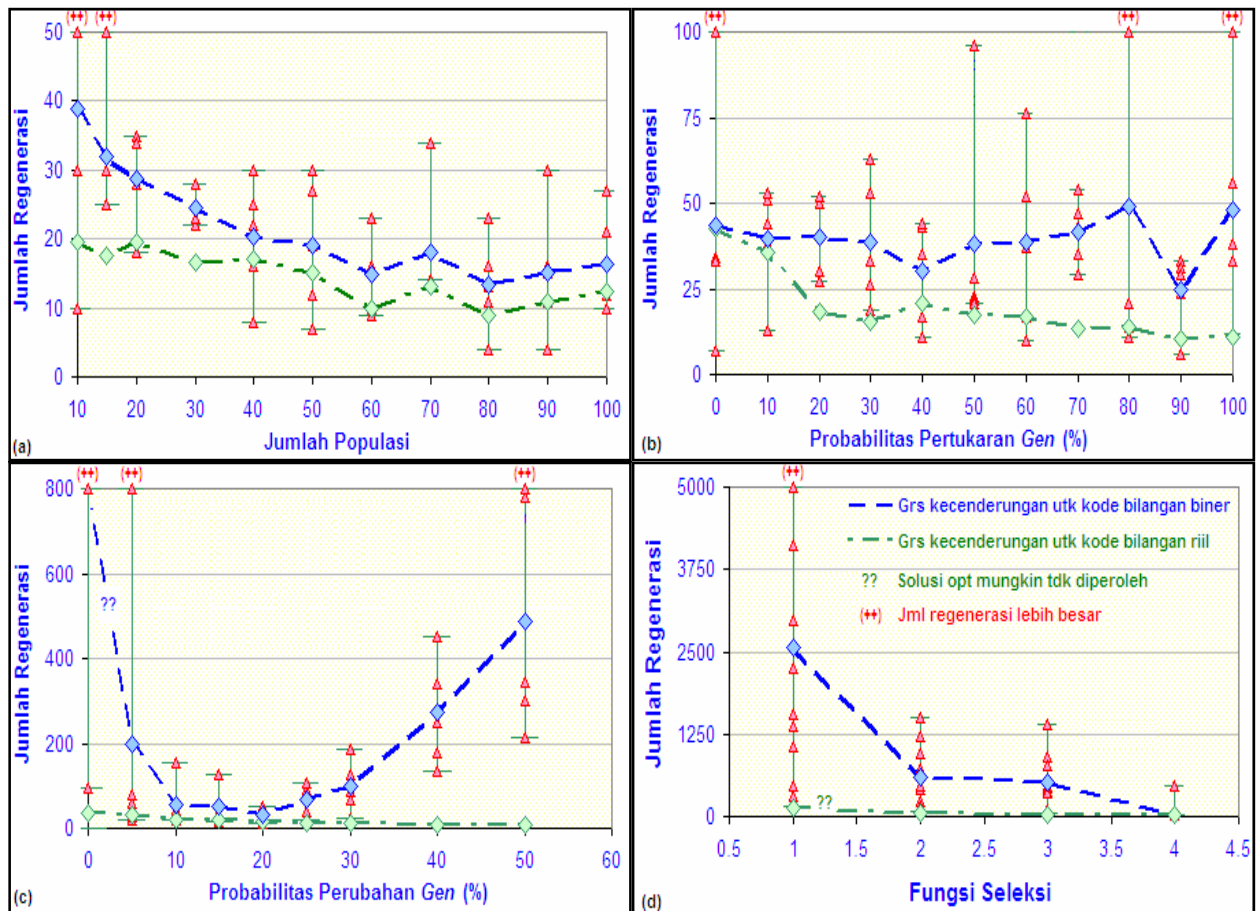
### 5.3. Perhitungan dengan algoritma genetika – kode solusi sebagai bilangan riil

Perkiraan nilai *ITP* akibat pengaruh dari variasi nilai dari keempat parameter utama algoritma genetika, untuk kodifikasi solusi sebagai bilangan riil, juga diperlihatkan pada **Gambar 3**. Secara umum, algoritma genetika dengan kodifikasi solusi sebagai bilangan riil ini lebih efisien. Jumlah regenerasi yang diperlukan untuk menghasilkan solusi optimum sekitar 10 kali. Hasil ini cukup konsisten dengan hasil yang diperoleh dari proses konvergensi yang menggunakan teknik interpolasi linier yang memerlukan proses iterasi sebanyak 11 kali (lihat **Bab 5.1**).

Untuk kodifikasi solusi sebagai bilangan riil, jumlah populasi, probabilitas pertukaran *gen* dan probabilitas perubahan *gen* yang makin besar cenderung dapat menghasilkan proses pencarian solusi optimum menjadi lebih efisien. Fungsi seleksi yang selalu terus mempertahankan individu terbaik pada setiap proses regenerasi (alternatif 2, 3 dan 4) juga membuat algoritma genetika efisien.

Sedangkan, fungsi seleksi random (alternatif 1) seringkali menyebabkan kegagalan dalam proses pencarian solusi optimum. Proses regenerasi dapat





**Gambar 3. Jumlah regenerasi yang dilakukan untuk berbagai variasi parameter utama algoritma genetika, yaitu :**

**(a) jumlah populasi (b) probabilitas pertukaran gen (c) probabilitas perubahan gen dan (d) fungsi seleksi**

terperangkap pada individu yang semuanya sama. Dalam kondisi demikian, proses pertukaran *gen* pada **Persamaan (4)** dan perubahan *gen* pada **Persamaan (5)** menjadi tidak berguna, karena individu turunan tidak akan pernah berbeda dengan individu induknya.

Dari **Gambar 3**, nilai parameter utama algoritma genetika yang dapat menjadikan proses pencarian nilai *ITP* efisien adalah:

Jumlah populasi = 100  
 Tingkat probabilitas pertukaran *gen* = 100 %  
 Tingkat probabilitas perubahan *gen* = 50 %  
 Kriteria seleksi : fungsi seleksi unggul (alternatif 4)

Dari analisis lebih lanjut terhadap keempat nilai parameter utama algoritma genetika tersebut diketahui, bahwa jumlah populasi merupakan parameter yang tidak begitu signifikan. Oleh karena itu, jumlah populasi seharusnya cukup terdiri dari 10 individu saja. Pengurangan jumlah populasi dalam generasi

yang dianalisis tentunya juga dapat membuat proses pengoperasian komputer lebih cepat.

## 6. Kesimpulan

1. Untuk contoh kasus pencarian nilai *ITP* dimana uji terminasi dapat dirumuskan, algoritma genetika meskipun didasarkan pada pendekatan *stochastic* umumnya dapat memberikan solusi yang eksak, yang persis sama dengan yang dihasilkan dari metoda konvergensi yang menggunakan teknik interpolasi linier. Sifat *stochastic* dari algoritma genetika hanya berpengaruh pada jumlah regenerasi yang harus dilakukan untuk mencapai solusi tersebut.
2. Algoritma genetika yang menggunakan kodifikasi solusi sebagai bilangan riil dapat memberikan hasil yang cenderung lebih efisien dibandingkan dengan yang menggunakan kode bilangan biner. Hal ini diperoleh khususnya karena operator genetika direksional yang diaplikasikan pada kode bilangan

riil menyertakan ketentuan tentang solusi optimum dalam modelnya (*meta heuristic model*).

3. Secara umum, dalam proses pencarian nilai *ITP*, nilai parameter utama algoritma genetika yang paling sesuai untuk kode bilangan riil dapat diringkaskan sebagai berikut. Jumlah populasi = 10 individu; probabilitas pertukaran *gen* = 100%; probabilitas perubahan *gen* = 50%, dan kriteria seleksi menggunakan fungsi seleksi unggul. Setelah melewati proses validasi, jumlah populasi ternyata merupakan parameter algoritma genetika yang tidak begitu signifikan.

### **Daftar Pustaka**

- AASHTO, 1993, “*AASHTO Guide for Design of Pavement Structures*”, Washington DC, USA.
- Gen, M., Cheng, R., 1997, “*Genetic Algorithms and Engineering Design*”, John Wiley and Sons, Inc., USA.
- Kosasih, D., 2005, “*Petunjuk Pengoperasian Program Genetika*”, Departemen Teknik Sipil, ITB, Bandung.
- Kosasih, D., Rinaldo, 2005, “*Analisis Aplikasi Algoritma Genetika Untuk Pencarian Nilai Fungsi Maksimum*”, Prosiding Temu Ilmiah Dosen 2005, Fakultas Teknik, UNTAR, Jakarta.